Indian Journal of Science and Technology, *Vol 9(13), DOI: 10.17485/ijst/2016/v9i13/71360, April 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Searching for an Unknown Edge in the Graph and its Tight Complexity Bounds

Abbas Cheraghi*

Department of Mathematics and Statistics, Khansar, University of Isfahan, Isfahan, Iran; a.cheraghi@khn.ui.ac.ir

Abstract

Assume that H is a graph and *c(R,H)* is the number of tests needed by an algorithm *R* in the worst case to find unknown edge *e** of H. The aim is to set $c(H) = min_R\, c(R,H)$. In this paper, presented a straightforward proof of the tight lower bound on *c(H)* for a special class of graphs. This explicit new bound based on the combinatorial object might be used for key distribution algorithm. Moreover, presented a lower bound on *c(H)* and show this bound for the complexity is tight.

Keywords: Combinatorial Search, Graph Searching, Group Testing, Key Distribution, Unknown Edge

1. Introduction

Numerous studies have been conducted about Graph searching[1,2,3,4,5], and it is, in fact, a subset of the bigger taxonomy of DNA application, blood testing, deceptive tests, finding patients about which a set of studies have been conducted (e.g., the book[1]). It was first mentioned by Parsons[6] and by Petrov[7] separately, and the first definition is what is recently called edge-searching. In the present setting, a group of experimenter intends to capture a floating edge along the edges of a graph. This floating edge move on time, pursue experimenter and is aware of the all of the moving of the experimenter, since the experimenter might not find it until they catch him, i.e., when this special edge is caught and cannot run anywhere. On the other hands the floating edge float on its graph as much as possible. An edge determines by shifting a searcher from one endpoint to another, and a vertex is cleared when a searcher is inserted on it. For example finding an edge in a simple graph without any multiple edges is equal to finding two endpoints of floating edge, so in a basic version of edge searching, we usually try to find endpoints vertices instead of its edge. The problem, however, is to know about the minimum number of examinations that might ensure the capturing of the floating edge called the edge search number of the graph.

One of the first applications of group testing problems was happening in World War II when a group of soldiers needs blood testing for determining syphilitic patients. But one by one blood testing needs a lot of cost and time, so the best idea was mixing the types of blood of a subgroup of soldiers and doing the test on it. If the answer to the test is negative it means that all the soldiers in that subgroup are safe and if not they must select another subgroup of soldiers more and more to find all the patients. The main goal is finding a minimum of blood testing in the worst case. The simple case of this group testing is when the number of patients is two of *n* soldiers. We model this kind by graph theory and called classical group testing.

The edge searching problem on graphs is, in fact, a development of the classical group testing problem. Suppose that we have a graph $H = (V, E)$ with vertex set V and edge set E. Let H_S indicate the subgraph of H induced by the set S of vertices. What we should do is to reach a subset $D \subseteq E$ of faulty edges with the smallest possible number of edge tests in which an edge test takes an arbitrary subset $S \subseteq V$ and see if the subgraph H_S has a defective edge or not. Actually, the main goal is to calculate a minimum number of induced subgraphs for finding a special edge in edge set of the graph.

Chang and Hwang[8] first introduced the problem of recognizing two defective vertices in a complete bipartite

**Author for correspondence*

graph, as a subgroup of patients in syphilitic patient's story. This problem might be regarded as a specific group testing problem of searching for a single edge on graphs. Aigner[9] was the first who offered the edge testing problem for a general graph and drew attention to it. In this paper, we offer a simple proof of the tight lower bound on the number of tests in group testing problem for searching an edge in graphs.

In what follows, two issues are addressed: One of the most important subjects of cryptography protocol called key distribution and the second one is the general case of edge searching problem called non-adaptive group testing algorithms.

Another interesting subject on cryptography is digital fingerprinting. For a combinatorial model of this subject, Boneh and Shaw[10] defined Frame-proof codes as an executive of it preventing an alliance of a specific size c from framing a user, not in the alliance. Later on, Stinson and Wei[11] offered a dual mode of the problem on the basis of specific kinds of set systems. Two new definitions, *(i, j)*-Cover-free family and *(i, j)*-disjunct system are used[12] to make *(i, j)*-key distribution algorithm and a non-adaptive group testing algorithm respectively; as they are dual incidence structures. Let's first introduce the related terminology about set systems.

Definition 1.1 A pair *(Y, y)* in which *Y* is a group of factors referred to as points and *y* is a family of subsets of *Y* whatsoever is called a set system. Every subset of *Y* is referred to as blocks.

Definition 1.2 Let *i* and *j* be positive integers and *(Y, y)* be a set system. This set system is called an *(i, j)*-cover-free family if for any two subsets C_1, C_2 of *y*, in which $|C_1| \le I$, $|C_2| \le j$ and $C_1 \cap C_2 = \varnothing$, we have $\bigcap_{B \in C_1} B \not\subset \bigcup_{B \in C_2} B$. When $|Y| = v$ and $|B| = y$ *we show an (i, j)*-cover-free family *(Y, y)*, with the notation *(i, j)*-CFF*(v, y)* briefly.

Symmetric cryptosystem used for large plain-text, but it needs a secure channel for establishing a common key between the sender and receiver as well. There are several protocols for the key establishment's problem. In the following definition explained the concept of establishing key between the participants of a private virtual meeting.

Definition 1.3 Assume two positive integers *v* and *y*. An *(i, j)*-key distribution algorithm is a way of spreading a collection of *v* keys to a group of *y* participants, in a way that each subgroup of *i* participants might plan a meeting keys by integrating their keys in common. Any meeting key, thus planned needs to be safe against a subset of an alliance of size at most *j*. We employ the notation *(i, j)*-KDP *(y, v)* for an *(i, j)*-key distribution algorithm as an abbreviation.

Actually, in an every private virtual meeting, the main goal is spreading a collection of keys between the participants in such away every qualified subset of participants recover a common key whereas no forbidden subset of participant wrest it. As a motivation, it is illuminated a method that key distribution algorithms result from cover-free families.

Let *(Y, y)* be an *(i, j)*-CFF *(v, y)*, such that $i \ge 2$. For every $x \in Y$, assume k_x be a key, picked randomly from an enough large finite field. Assume also that there is a group of *y* users, denoted u_B $(B \in y)$, and every user u_B is handed the keys k_x $(x \in B)$. Suppose *C* is a subgroup of *i* users. Hence, for any other disjoint alliance *D* of size at most *j*, there is a key held by each user of *C* and by no user of *D*. If assume that the meeting key k_C is

$$k_C = \sum_{\{x : x \in B \ \text{for all} \ B \subseteq C\}} k_x,$$

Therefore, every user of *C* might estimate the meeting key k_C, however, the value of k_C cannot be calculated by no more than *j* user in alliance *D*.

Definition 1.4 Let *i* and *j* be positive integers and *(Y, y)* be a set system. This set system is called an *(i,j)-disjunct system* provide if for any $P, Q \subseteq Y$ in a way that $|P| \le i$, $|Q| \le j$ and $P \cup Q = \varnothing$, there is a $B \in y$ in a way that $P \subseteq B$ and $Q \cup B = \varnothing$. When $|Y| = v$ and $|B| = y$ *we show an (i, j)*-disjunct system *(Y, y)*, with the notation *(i, j)*-DS*(v, y)* briefly.

Cover-free families and disjunct systems are, in essence, *dual* incidence structures. One of the group testing modes in which all the tests are known ahead of time is called non-adaptive group testing algorithm. Non-adaptive group testing algorithms are defined here informally.

Definition 1.5 Assume that *Y* is a test tube rack of *v* blood test tubes that should be tested as a positive or negative test. Assume also that *y* is a family of subsets of the tubes on *Y*, where each $B \in y$ is a *group* of blood test types that should be mixed and tested. The testing process has the feature that a group involves, at least, one positive sample, so the test result for the group is positive and otherwise the result is negative and concludes that all of the samples contain in B are safe. Let the testing procedure permits the identification of the positive blood test types if the number positive blood test types are at most *d* and also all the mixed blood test types are known ahead of time, then

this algorithm is referred to as a non-adaptive group testing algorithm and is denoted by *d*-NAGTA *(v, y)* briefly.

The phrase "non-adaptive" indicates the mixed blood test types implemented are fixed beforehand and do not hinge on the results of earlier tests.

For making a group testing algorithms we can employ disjunct systems as follows. Assume that *(Y, y)* is a *(1, d)*-DS *(v, y)*, in which *Y* is the test tube rack of *v* blood test types and *y* is the family of *y* groups. It is; then, obvious that the blood test types take place in no group that test positive are in fact the negative blood test types. Thus, the positive blood test types are recognized by this testing process, a *d*-NAGTA *(v, y)* result.

2. Tight Lower Bound on c(H)

In this section, we study a kind of group testing problem for two damaged elements by graph-theoretic properties. Let a collection *V* of *n* distinct elements has exactly two damaged elements. We want to interpret the search domain *V* as the vertex set of the graph *H* and search for two damaged elements adjacent each other, i.e., an unfamiliar edge *e** in the edge set *E* of *H*. The main goal is to find the unfamiliar damaged edge by continually selecting subgroups *U* of *V* and querying questions "is somewhat of the elements of *U* an element of *e**?". We called "*test*" to this kind of query and "*test set*" to subgroup *U*, as well.

Assume that H is a graph and *c(R,H)* is the number of tests needed by an algorithm *R* in the worst case to find unknown edge *e** of H. The main goal is to compute $c(H) = min_R\ c(R,H)$. Determining *c(H)* in general is very hard and is an open problem yet. There are various studies about the lower and upper bound for this term. The absorbed researcher will find more specifics around group testing problems in[13-17]. We begin with a simple but useful result. The next simple lemma is useful in future results.

Theorem 2.1 Let H be a graph and *F* is its subgraph, then we have, c*(F)* ≤ *c(H)*.

Proof. All of the tests and tests set in *F* are also in the *H* and so the minimal tests set on *F* is less or equal than the minimum tests set in *H*.

Let us make the first observation on this problem. Suppose in the course of our algorithm we have arrived at a graph *H(V, E)* in which the unknown edge *e** lies. The next test $A \subseteq V$ splits *H* into two parts H_1 and H_0 (Figure 1). If the answer is "yes", then *e** lies in H_A or in $H_{A,\ V\text{-}A}$, whereas if the answer is "no", *e** lies in $H_{V\text{-}A}$ Thus

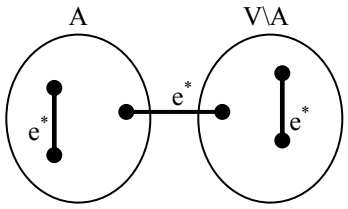

Figure 1. Splits G into two parts G1 and G0.

$H_1 = H_A \cap H_{A,\ V\text{-}A}$, $H_0 = H_{V\text{-}A}$, in fact, H_0 is an induced subgraph of graph *H*. So any test corresponds to a partition of *H* literally is an induced subgraph of *H*. It is convenient to interchange the roles of "yes" and "no" which obviously has no influence on *c(H)*. That is, after any test $A \subseteq V$ we obtain as feedback $e^* \in H_1 = H_A$ or $e^* \in H_0 = H_{A,V\text{-}A} \cap H_{V\text{-}A}$. From now on, we will always consider this latter version of our problem. There is a famous information theoretic bound[9] on *c(H)*.

Theorem 2.2 Assume that *H = (V, E)* is a graph. The lower bound on *c(H)* is $c(H) \geq \lceil \log_2 |E| \rceil$.

In this paper, we present a straightforward proof of the tight lower bound on the *c(H)* in group testing problem for searching an edge in graphs that is sharpened than the famous information theoretic bound on *c(H)*.

Let us first discuss some necessary results that we need for the proof of this new bound.

Theorem 2.3 Assume that *H* is a graph with vertex set *V* and edge set *E* such that $|E| + |V| > 2^k$ for $k \geq 4$, then $c(H) \geq k+1$.

Proof. We prove $c(H) \geq k+1$ holds by induction on *k*. For $k = 4$, the only interesting case is $|V| = 6$. M. Aigner[9] prove that $c(K_6) = 5$, thus $c(H) \geq k+1$. Let *A* be the first test set, so *H* splits into two parts induced subgraph $H_1 = H_A$ and the remainder $H_0 = H_{A,V\text{-}A} \cap H_{V\text{-}A}$.

If $|E(H_1)|+|V(H_1)| > 2^{k-1}$, so according to the induction $c(H_1) \geq k$. Because of the first question we have $c(H) \geq k+1$. Otherwise, if $|E(H_1)|+|V(H_1)| \leq 2^{k-1}$ by the definition of H_1 and H_0 we have $|E(H_0)|+|V(H_0)| > 2^{k-1}$ and again follows by induction applied to H_0 we have $c(H0) \geq k$. Then because of the first question "*A*", we have $c(H) \geq k+1$.

A complete graph with *n* nodes is a simple graph such that every pair of distinct *n* nodes is adjacent with a unique edge and show by the notation K_n.

Theorem 2.4 Assume that K_n is a complete graph. Then for every *k* greater than 3 such that $\binom{n+1}{2} > 2^k$ we have the lower bound $c(K_n) \geq k+1$.

Proof. Suppose $|E(K_n)|$ and $|V(K_n)|$ are the size of the edge set and vertex set of K_n with n vertices, respectively. Then

$$|\mathrm{E}(\mathrm{K_n})|+|\mathrm{V}(\mathrm{K_n})|=\binom{n}{2}+n=\binom{n+1}{2}$$

Therefore by assumption, the assertion is trivial.

It is obvious there exists an integer k with $\binom{n}{2}\le 2^k < \binom{n+1}{2}$. The preceding proposition shows the next result that is a special case of theorem 2.2 for a large n.

Theorem 2.5 Let K_n is a complete graph. Then we have $c(K_n)\ge\left\lceil \log_2\binom{n}{2}\right\rceil$.

As mentioned before the group testing problem is dual of key distribution pattern and the exact value of c(H) actually is the minimum number of keys must be established between the users, so this explicit new bound based on a combinatorial object can be used for key distribution algorithm.

Theorem 2.6 Assume that H is a graph with vertex set V and edge set E. Then $c(H)\ge\lceil \log_2|E|+|V|\rceil$.

Proof. It is obvious there exists an integer k with $2^{k+1}\ge |E|+|V|>2^k$ so $\lceil \log_2|E|+|V|\rceil=k+1$, then the assertion is trivial by Theorem 2.3.

The next example indicates that the lower bound presented in Theorem 2.6. is sharp.

Example 2.7 Assume that H is a complete graph with $n = 2^l$, in which $l\ge 3$. With this assumption, we conclude that $\binom{n}{2}=\frac{2^l(2^l-1)}{2}=2^{2l-1}-2^{l-1}<2^{2l-1}$ and $\binom{n+1}{2}=\frac{2^l(2^l+1)}{2}=2^{2l-1}+2^{l-1}>2^{2l-1}$, hence $\binom{2^l}{2}<2^{2l-1}<\binom{2^l+1}{2}$, $l\ge 3$. Our result implies $\mathrm{c(K_n)}\ge\left\lceil \log\binom{2^l+1}{2}\right\rceil\ge 2l$.

On the other hands by induction on l, we prove that $c(K_n)\le 2\,l$ for $n = 2^l$ and all $l\ge 1$. For $l = 1$ this is obvious. We split the vertex set of K_n into two equal sized parts A and B with $|A| = |B| = 2^{l-1}$. As first test set we take A and as the second set, we take B. After these two tests, we know that the unknown edge e^* lies in $H_A = K_{n/2}$ or in $H_B = K_{n/2}$ or in $H_{A,B} = K_{n/2,n/2}$. For the first two possibilities by induction, we have $c(H_A) < 2l\text{-}2$, $c(H_B) < 2l\text{-}2$, and thus $c(K_n)\le 2l$. If e^* is in $H_{A,B}$, then we know that one end vertex u is in A while the other end vertex v is in B. By the usual halving method, we can identify u with $l\text{-}1$ tests on A, and similarly for v. Thus we again obtain $c(H_{A,B})\le 2l\text{-}2$, i.e. $c(K_n) < 2l$. We have thus proved $c(K_n) = 2l$ *for* $l\ge 3$.

3. Acknowledgments

The author is grateful to the referee for helpful suggestions. This work was supported by the Research Grant Khansar-CMC-007.

4. References

1. Alpern S, Gal S. The theory of search games and rendezvous. International Series in Operations Research and Management Science. Kluwer Academic Publishers, Boston, MA. 2003; 55.
2. Golovach PA, Heggernes P, Mihai R. Edge search number of cographs. Discrete Applied Mathematics. 2012; 160(6):734–43.
3. Mazoit F, Nisse N. Monotonicity of non-deterministic graph searching. Theoretical Computer Science. 2008; 399(3):169–78.
4. Megiddo N, Hakimi SL, Garey MR, Johnson DS, Papadimitriou CH. The complexity of searching a graph. Journal of the ACM. 1988; 35(1):18–44.
5. Yang B, Zhang R, Cao Y. Searching cycle-disjoint graphs. Proceedings of First International Conference on COCOA, Xian China. 2007. p.32–43.
6. Parsons TD. Pursuit-evasion in a graph. Theory and Applications of Graphs, Springer-Verlag : Heidelberg, 1976. p.426–41.
7. Petrov NN. A problem of pursuit in the absence of information on the pursued. Differentsial'nye Uravneniya, 1982; 18(8):1345–52.
8. Chang GJ, Hwang FK. A group testing problem on two disjoint sets. SIAM Journal on Algebraic and Discrete Methods. 1981; 2(1):35–38.
9. Aigner M. Combinatorial Search, Wiley-Teubner Series in Computer Science, Wiley, New York. 1988. p.128–45.
10. Boneh D, Shaw J. Collusion secure_finger printing for digital data. IEEE Transactions on Information Theory. 1998; 44(5):1897–1905.
11. Stinson DR, Van Trung T, Wei R. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. Journal of Statististcal Planning Inference. 2000; 86(2):595–17.

12. Stinson DR, Wei R. Combinatorial properties and constructions of traceability schemes and frameproof codes. SIAM Journal of Discrete Mathematics. 1998; 11(1):41–53.
13. Alswede R, Wegener I, Suchprobleme, Teubner, Stuttgart; 1979.
14. Dorfman R. The detection of defective members of large population. Annals of Mathematicals. Statistics. 1943; 14(4):436-40.
15. Chandu PMSS, Sasikala T. Implementation of regression testing of test case prioritization. Indian Journal of Science and Technology. Apr 2015; 18(S8):290-3.
16. Du DZ, Hwang FK. Combinatorial group testing and its applications. Word Scientific, Singapore; 1993.
17. Du DZ, Hwang FK. Pooling designs and non-adaptive group testing: Important tools for DNA sequencing. Word Scientific: New Jersey. 2006. p.1-20.